可程式設計安裝程式

目錄

可程式設計安裝程式

欄位	價值
產品	可程式設計安裝程式
檔案型別	技術白皮書 — 架構、實施和成果
主要受眾	解決方案架構師、平台工程師、DevOps/SRE、交付主管
次級受眾	工程管理、安全審查員、專案經理

摘要

可程式設計安裝器是一個規範驅動的自動化平台，用於在企業Linux（RHEL系列）和相關基礎設施（VMware vCenter、OpenShift、KVM、空隙Kubernetes）上部署和操作思科軟體堆疊(包括網路服務協調器(NSO)、Crosswork網路控制器(CNC)、Crosswork資料網關(CDG)和業務流程自動化(BPA))。 系統將宣告性意圖（YAML規範和可選的指導性意圖生成）與執行(Ansible roles and playbook)分開，並使用Pythoncontrol計畫封裝對象、在長時間運行安裝之前驗證包、準備加密的機密以及協調驗證門。

本白皮書將介紹架構層、主資料流、實施模式（包括混合資料驅動的工件驗證模型）、部署模式（本地、容器化、聯機、空隙）以及驗證和日誌框架。對於交付和平台組織，該平台旨在儘早減少手動工作、表面錯誤配置和缺少二進位制檔案，並在保持特定於環境的引數化的同時，跨產品和拓撲實現自動化標準化。

關鍵詞：基礎設施自動化、宣告性部署、Ansible、YAML規範、氣隙封裝、人工驗證、Crosswork、NSO、CNC、CDG、BPA、驗證策略、DevOps。

如何閱讀此文檔

角色	建議的焦點
決策者/領導者	摘要；§1價值主張；§8效益和風險狀況；§10結論
解決方案架構師	§3-§6（體系結構、規格模型、實施、部署模式）
開發運營/SRE/交付工程師	§5-§7;附錄B;隨附內部白皮書附件
安全稽核員	§7安全性和合規狀況；%3.2中§信任邊界

1.價值主張

1.1交付問題

企業安裝多層網路自動化和協調產品是傳統的高接觸式方式:長的Runbook、許多手動步驟、站點之間的版本偏差，以及表面進入流程的故障(缺少NED、錯誤的OVA路徑、空隙映像集不完整)。 這種模式增加了成本，延長了更改視窗，並且使稽核更困難。

1.2可程式設計安裝方法

可程式設計安裝程式將安裝視為通過特定引數化的程序：拓撲、版本、平台選擇（vCenter與OpenShift與vanilla VM）、檔案路徑和授權。自動化可能具有潛在性，可在所有客戶之間重複，並帶有檢查的前端負載，因此「未就緒」是在群集或產品安裝之前一個快速、明確的結果。

1.3 「可程式設計」在此背景下意味著什麼

- 宣告：操作員說明應部署的內容；實戰手冊的實施。
- 資料驅動驗證：當模式穩定時，每個版本期望的對象從表和規則中匯出，而不是臨時指令碼。
- 策略控制品質：部署前和部署後驗證在分層策略下運行，具有結構化報告和可選的故障單整合。
- 多模式操作：首次使用者的互動式選單；CLI和凍結的二進位制檔案，用於辭彙/光碟形式的重複；用於標準化運行時映像的基於Docker的流。

2.系統背景

2.1行為者和環境

參與者/系統	角色
交付工程師	編寫或生成規格、運行打包、儲存庫準備、驗證、協調器和Ansible
安裝程式主機	Linux控制節點（本機或容器），帶Python、Ansible配置、磁碟用於工件
目標基礎設施	vCenter、OpenShift/KubeVirt或按規範的Vanilla VM
對象源	內部映象、授權佈局、軟體分發 — 特定於環境
下游系統	監控、變更管理、可選的JIRA工作流程

2.2信任邊界

1. 規格明示意圖；它們可以引用路徑和非機密引數。機密應流經Ansible Vaultworkflows，而不是可避免的純文字檔案規範欄位。
2. 物品存儲必須受到完整性保護；驗證重點是線上狀態和名稱與規範保持一致 — 在策略要求的情況下擴展組織控制（校驗和、帶簽名的捆綁包）。
3. 安裝程式到目標的SSH是高許可權路徑；安裝程式主機受到危害影響極大。強化和訪問控制是操作先決條件。

3.建築原則

1. 宣告性優先:YAML中的使用者意圖；自動化對其進行一致的解釋。
2. 規劃和執行分離：Python規劃、驗證和協調；Ansible執行基礎設施和產品步驟。
3. 可組合自動化：站點級行動手冊匯入重點行動手冊（預安裝、Kubernetes路徑、產品安裝）。
4. 漸進式披露：用於入職的互動式設定；用於高級自動化的標籤和指令碼。
5. 相同的代碼庫、多個運行時間：本地AlmaLinux/RHEL路徑和基於Docker的路徑共用一個儲存庫佈局。
6. 顯式氣隙支援：在連接的機器上封裝；轉讓套件組合；安裝前提條件並在公共網路上無運行時依賴關係進行部署。

4.邏輯架構

4.1層

層	責任
規格	拓撲、版本、平台、路徑、授權
控制平面	打包、捆綁包驗證、保險儲存幫助程式、驗證驅動程式、協調器CLI
自動化平面	主機準備、Kubernetes生命週期、產品安裝和第一天配置
專案	二進位制檔案、影象、圖表、OVA、tarball

4.2端到端資料流

1. 封裝流程：必要的工具將檔案下載或暫存到特定位置，可選擇為離線安裝生成可轉移的圓球。
2. 驗證流程：協調器解析規範，解析預期對象（包括平台過濾器和權利清單），並在安裝前報告就緒/丟失。
3. 部署流程：Spec plus保管庫（和可選的預驗證）根據從專案生成的清單驅動Ansible手冊。

4.3支援的產品（安裝程式範圍）

產品/套件組合
NSO
CNC
CDG
BPA
CNC + NSO

5.規格和意向模型

5.1使用者規格

規範是描述平台(例如vCenter、OCP、VM、KVM)、主機、具有版本的應用程式、拓撲（例如NSO CFS/RFS佈局）、entitlations（NED和附加軟體包）、用於OVA的andfilepaths、qcow2映像和應用層陷阱。

5.2通用片段

特定應用程式user_spec為CNC/CDG提供預設值和路徑回退。協調器的解析器將使用者規格視為可信來源，並在使用者金鑰缺失時使用通用規範條目。

5.3意圖生成

意圖生成器通過一組調查表、規則引擎(日期驅動邏輯)和到intent.yaml的模式支援對映。

6.執行深入探討

6.1部署協調器(`cx_deploy_orchestrator.py`)

協調器是指令碼化或interactivegenerate-intent、verify-bundle和installcoordination的單一入口。其設計是明顯混合的:

- **`ARTIFACT_DEFS`**:宣告每個應用程式的對象型別和命名模式(NSO安裝程式、NED簽名包、可選包；CNC OVA/qcow2/tier tarball;CDG影象；BPA圖表和氣隙影象標籤)。
- **`APP_CONFIG:`**

將CLI對映- appvalues(nso、crossworksuite、bpa)對映到規範資料夾名稱和預設目的檔名。

- **解析器/處理程式**:使用使用者規範和通用規範解析CNC/CDG路徑；custom **handlerscover非統一命名（例如TSDN/DLM）以及圖表和軌跡路徑的BPA版本格式**。

就緒性:AReportaggregates發現的工件;is_readyis true時，在規範分析後沒有丟失所需的檔案。該模組支援PyInstaller凍結的二進位制檔案viasys.frozenpath解析。

6.2必備項和打包工具(setup_cxinstaller_prereqs)

此元件為專案打包和主機先決條件安裝提供互動式menusandCLI模式：線上、氣隙或自動檢測；包括組合CNC_NSO的多應用程式套件。它填充Ansible和verify-bundle邏輯預期的對象樹。

6.3 Ansible自動化平面

組合：此說明堆疊的多重追蹤性質：它匯入不同的Kubernetes載入程式路徑 — 反映不同的產品以不同的Kubernetes載入程式路徑為目標。

角色（代表系列）：preinstall（SELinux、防火牆、SSH、登錄檔幫助程式）、k8s_install/rke2、deploy_nso、deploy_cnc、deploy_cdg、deploy_bpa、postinstall、解除安裝和證書更新幫助程式。所有權和測試是最好的管理暴力邊界；巢狀任務資料夾實現子工作流（例如NSO L3 HA）。

6.4驗證檢查框架(validation_checks/)

框架提供Archierarchical策略控制(global → app → stage → individual check)、auto-discoveryof checks、enhanced reporting to structured logs和可選JIRAintegration。操作員按照用於安裝的相同規範運行部署前或部署後階段，使自動化與適合企業變更規範的品質網關保持一致。

典型分支上的指示性刻度：按照BPA和NSO的thirtychecks順序(計數應通過簽出時的make list-validation-checkson確認)。

6.5保險庫保密管理器(scripts/vault_secrets_manager.py)

從規範中匯出所需的儲存庫變數，在策略下提示或接受密碼，並為Ansible生成emitsencryptedgroup_vars/all_secrets.yamlplus儲存庫密碼檔案，從而減少在行動手冊中臨時加密的嵌入。

7.部署模式和運行時間

模式	摘要
本機(AlmaLinux / RHEL)	SetPYTHONPATHandANSIBLE_CONFIG;根據產品指南運行包裝、保險儲存、驗證、協調器和手冊
基於Docker的安裝程式	scripts/setup_installer.shandscripts/start_installer.shwith大型專案的主機裝載；
氣隙	包裝在已連線的機器上；傳輸捆綁包；目標提取；隨 — airgap安裝
macOS捆綁包建立	使用epython3 ./setup_cxinstaller_prereqs.pyon Mac準備捆綁包；目標部署仍按專案文檔面向Linux

---

8.安全性、驗證和可觀察性

8.1安全狀態（設計意圖）

- 秘密：優選Ansible Vault加密組變數；在合適的情況下使用保險儲存管理器嚴格模式。
- 安裝程式主機：視為高信任控制平面；限制訪問和監控。
- 對象：保護採集通道；組織過程可以通過加密驗證來增強驗證捆綁。
- 日誌記錄：應用程式和Ansible日誌未部署/日誌/

8.2驗證為操作門

部署前呼叫示例：

```
cd /opt/cx-installer
python3 validation_checks/run_validation_checks.py -t pre -s specification/your_spec.yaml
```

使用可選標誌：

- -p <policy_file> — 使用自定義驗證策略(默認值為 validation_checks/validation_policies/default.yaml)
- -a <app> — 將檢查限制到特定應用(小寫，例如cnc、nso、bpa、cdg)
- –report-file <path> — 寫入獨立JSON預檢查報告

8.3發佈規則

這是一種內部資源調配模式，對於更多的思科應用安裝，遵循相同的原則可劃分儲存庫。

9.惠益和成果

主題	結果
時間和辛勞	減少手動步驟；在驗證捆綁包和驗證階段檢測到故障，而不是在Ansible或產品安裝程式中後期檢測到故障
一致性	跨服務共用的架構、角色和專案佈局減少了「snowflake」差異
斷開連線的操作	有文檔記錄的捆綁傳輸支援受管制網路，無需運行時下載
治理	結構化驗證報告和可選的JIRA掛鉤支援變更記錄和跟進
可擴充性	清除擴展點：ARTIFACT_DEFS、處理程式、新角色/行動手冊、意圖架構

量化指標（安裝持續時間、缺陷率）特定於組織;團隊應參照可比較拓撲上的舊版runbook。

10.可擴充性和維護

10.1新增對象型別

1. ExtendARTIFACT_DEFS(如果需要，還會新增標籤)，包括x_deploy_orchestrator.py。
2. 當命名無法單獨由模式捕獲時，新增acustom控制代碼。
3. 自動下載時，更新setup_cxinstaller_preresq中的打包邏輯。

10.2新增可能的行為

優先使用位於聚合角色內的新任務；邊界清晰時引入新節點。Wire playbooks viaimport_playbookor已記錄的入門手冊。Keepsafe defaultsingroup_vars/vars。

10.3意圖生成器的演變

更新YAML模式下目的生成器/模式/和chatbot輸入；確保生成的檔案與APP_CONFIG預期的檔名匹配。

10.4路線圖注意事項（示例）

- 更深的BOM或影象簽名驗證整合。
- 擴展了CNC/CDG方案的驗證範圍。
- 用於規範連結和Ansible語法檢查的CI引用。

11.結論

CX可程式設計安裝程序組合了宣告性規範、Python控制規劃（用於打包和驗證）和Ansible automation規劃（用於跨各種基礎架構和連線模型的可擴展、可重複地部署Crosswork相關產品）。其架構有意將intinterexecution、appliesdata-drivenartifact expectations(實際應用)以及embedsvalidation and vaultworkflows（適用於企業交付的工作流程）分開。有關完整的操作附件（攻略表、故障排除矩陣、連線矩陣和擴展命令參考），請參閱配套的內部白皮書。

參考資料和文檔結構圖

檔案	Path
操作員指南	README.md
發行攻略	RELEASE_GUIDE.md
內部架構附件	docs/CX_INSTALLER_TECHNICAL_WHITE_PAPER_INTERNAL.md
Docker（線上/氣隙/使用）	SETUP_ONLINE_DOCKER.md,SETUP_AIRGAPPED_DOCKER.md,USAGE_DOCKER.md
驗證框架	docs/validation_checks/README.md
儲存庫管理器	docs/scripts/VAULT_SECRETS_MANAGER.md
產品指南	docs/nso.md、docs/bpa.md、docs/CNC_VCENTER_DEPLOYMENT_GUIDE.md、 docs/CNC_OCP_DEPLOYMENT_GUIDE.md、docs/CNC_NSO_DEPLOYMENT_GUIDE.md
意圖生成器	intent-generator/README.md
聊天機器人/規則流概述	docs/HowItWorks.md

附錄A — 儲存庫佈局 — https://wwwin-github.cisco.com/CX-SAO-TOOLS/cx-installer（摘要）

```
cx-installer/
├── ansible_playbooks/     # ansible.cfg, files/, group_vars/, playbooks/, roles/, vars/
├── apps/                  # App-specific supporting content
├── deploy/                # Python deploy helpers, logging utilities
├── docs/                  # Technical documentation
├── intent-generator/      # Chatbot, rule engine, schemas, output/
├── scripts/               # Docker setup/start, vault_secrets_manager.py, ...
├── specification/         # User specs, samples, common fragments
├── validation_checks/     # Policies, runners, reports
├── cx_deploy_orchestrator.py
├── setup_cxinstaller_prereqs*
├── requirements.txt
└── README.md
```

安裝後專案焦點（典型）：ansible_playbooks/files/artifaces/、files/bin/、files/charts/、files/images/。

附錄B — Orchestrator CLI（摘要）

指令碼:cx_deploy_orchestrator.py

引數	說明
– 應用/-a	nso|crossworksuite|bpa
– 規格/秒	YAML規範的路徑
– 步驟	generate-intent|verify-bundle|install
-verify-only	驗證套件組合；未就緒時退出非零
– 乾跑	在支援的位置進行乾式運行
–list-specs	列出已知規格

環境（典型會話）：

```
export PYTHONPATH=$(pwd)
export ANSIBLE_CONFIG=$(pwd)/ansible_playbooks/ansible.cfg
```

附錄C — 辭彙表

字詞	定義
規格	YAML使用者規範：平台、應用、拓撲、路徑、授權
意圖	從意圖生成器或手寫的等效項進行規範化YAML
套件組合	用於氣隙傳輸的打包安裝程式樹（通常為tarball）
協調器	cx_deploy_orchestrator.py — 驗證/意圖/安裝協調
專案驗證	檔案系統檢查每個規範中是否存在所需的二進位制檔案/映像
儲存庫	Ansible Vault-encrypted variable file for secrets
NED	網路元素驅動程式套件(NSO)
CFS/RFS	NSO群集轉發器/冗餘轉發器拓撲概念
氣隙	沒有安裝程式時訪問包下載終結點的環境

文檔修訂歷史記錄

版本	日期	備註
1.0	2026-03-27	初步發佈就緒技術白皮書（可程式設計安裝程式成幀）

關於此翻譯

思科已使用電腦和人工技術翻譯本文件，讓全世界的使用者能夠以自己的語言理解支援內容。請注意，即使是最佳機器翻譯，也不如專業譯者翻譯的內容準確。Cisco Systems, Inc. 對這些翻譯的準確度概不負責，並建議一律查看原始英文文件（提供連結）。